

"Express Mail" mailing label number EL485651283US
Date of Deposit December 19, 2000

Atty Docket No. 00P7662US01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

This is a U.S. Patent Application for:

**TITLE: SYSTEM AND METHOD FOR ANALYZING AND GENERATING
SUPPLEMENTARY SERVICE DATA UNITS IN PACKET BASED
MULTIMEDIA COMMUNICATIONS SYSTEMS**

Inventor #1: Florian Trinkwalder
Address: 534 Giuffrida Avenue, #A, San Jose, California 95123
Citizenship: Germany

Inventor #2: Mark E. Clark
Address: 2986 Little Rock Drive, San Jose, California 95133
Citizenship: USA

Inventor #3: Mark Skrzynski
Address: 1800 42nd Avenue, Capitola, California 95010
Citizenship: USA

SYSTEM AND METHOD FOR ANALYZING AND GENERATING SUPPLEMENTARY SERVICE DATA UNITS IN PACKET BASED MULTIMEDIA COMMUNICATIONS SYSTEMS

5 CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of commonly owned U.S. Patent Application Serial No. 09/586,408, filed June 1, 2000, and titled "System and Method for Sending Messages Defined by an ASN.1 Structure from an Application to the Signaling Entity and Vice Versa Via a Tree
10 Structured Text String," which is hereby incorporated by reference in its entirety as if fully set forth herein.

RESERVATION OF COPYRIGHT

A portion of the disclosure of this patent document contains material
15 which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the U.S. Patent and Trademark Office patent file or records available to the public, but otherwise reserves all
20 copyright rights whatsoever.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to telecommunication systems and, particularly, to a system and method for sending messages from an
25 application to signaling entities and, for particular embodiments, the sending of APDUs between H.323/H.450 endpoints.

DESCRIPTION OF THE RELATED ART

The Abstract Syntax Notation One (ASN.1), described in the
30 International Telecommunications Union (ITU) X.680 and X.691 specifications, defines a data structure protocol for describing messages to be exchanged between distributed computer systems. ASN.1 defines data units independently of system architecture. Because of this, ASN.1 is used to generate programming language code that forms the core of a wide variety of

messaging systems applications, including ITU-T Recommendation H.323.

The ITU-T Recommendation H.323 is a group of specifications defining the operation of a multimedia communication system over packet networks.

The ITU-T Recommendation H.323 specifies H.245 control signaling for negotiation of media channel usage, Q.931 (H.225.0) for call signaling and call setup, H.225.0 Registration, Admission, and Status (RAS), H.450 for supplementary services, and RTP/RTCP for sequencing audio and video packets. An exemplary system implementing the Recommendation H.323 is the HiPath™ 5500 system, available from Siemens Information and

Communication Networks, Inc.

H.323-based systems use ASN.1 coded messages and protocol state machines for describing the application protocol data units (APDU) or packets of data used for signaling between H.323 endpoints, servers, gateways, and gatekeepers.

The processing of H.323 ASN.1 messages is typically implemented in a protocol stack, and its functions are accessed by application programs through application programming interfaces (API). The software primitives of the APIs are typically written with programming languages, such as C, C++, and Java, and all the associated parameters are expressed and declared in the corresponding language.

Whenever new features, such as new parameters or completely new supplementary services are added to the protocol stack of a software product, such as an H.323 application, software changes must be done in three layers: new functional entities must be added to the protocol stack; new parameters or programming primitives must be added to the API; and applications and their user interfaces must be changed or enhanced to allow user access to the new features. Each operation carried out by the new feature must be callable via an API that must be explicitly exported to the application. The changes must be made at the source code level, and the new software system must be recompiled and linked together and loaded to the customer's target system.

Thus, changes and additions to the software require replacement of

the software in the target system. While software can be downloaded remotely, an interruption of the target system's operation occurs, which can hinder the simple "plugging in" of new features to the system.

Moreover, unless the API is modified, the application cannot access
5 new features. Thus, even relatively small changes and additions to the signaling protocols and related protocol stacks require relatively high development effort, since the API source code must be modified.

Therefore, there is a need for an improved method for updating the protocol stack of an API based software product. There is particularly a need
10 for an improved method for updating the processing of ASN.1 messages in an H.323 telecommunications system.

Further, as is known, H.225 messages are used for call signaling to establish a connection between endpoints. Initially, an H.225.0 RAS (registration, admission, status) channel is established between an endpoint
15 and a gatekeeper (in the gatekeeper-routed call model). After the RAS signaling channel is established, an H.225.0 call signaling channel is established between endpoints. H.225.0 call signaling messages include SETUP, ALERTING, CONNECT, RELEASE COMPLETE, and FACILITY. The user-user information element of an H.225 message can carry an H.450
20 APDU (application protocol data unit) for H.450 supplementary services. Such services are call-related services beyond the basic call and can include Call Forwarding, Call Transfer, Call Waiting, Message Waiting Indication, Conference, and the like.

As can be appreciated, newly implemented H.450 supplementary
25 service features must be tested with all possible normal and exceptional events. To perform these tests, various H.450 messages must be sent and received by the endpoints. As such, there is a need for a system and method for sending, receiving, and displaying such test messages.

30 SUMMARY OF THE INVENTION

These and other problems in the prior art are overcome in large part by a system and method according to the present invention. According to the

present invention, signaling messages and parameters that are defined by an ASN.1 structure are sent as a tree-structured text string from an application to a signaling entity (SE) and vice versa. As new signaling entities are added to support new features, the features can receive and send primitives without
5 the programmer having to change or expand the API.

According to an implementation of the present invention, application programming interfaces (APIs) are provided which implement a first recursive function for interpreting text strings representative of ASN.1 structures for communicating with an application program. A second recursive function
10 receives and interprets ASN.1 value trees from signaling entities (SE).

According to an implementation of the invention, an H.450 test client is provided. The H.450 test client can send and receive H.450 application protocol data units in any H.225 message. The test client allows for the setting of desired H.450 parameters and allows for the display of the H.450
15 message itself.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the invention is obtained when the following detailed description is considered in conjunction with the following drawings in
20 which:

FIG. 1 is a block diagram illustrating an implementation of the present invention;

FIG. 2 is a functional diagram illustrating an implementation of the present invention;

25 FIG. 3 is a flowchart illustrating an implementation of the present invention;

FIG. 4 is a functional diagram of an implementation of the present invention;

30 FIG. 5 is a flowchart illustrating operation of an implementation of the invention;

FIG. 6 is a graphical user interface according to an implementation of the present invention;

FIG. 7 is a diagram of another aspect of the graphical user interface of FIG. 6;

FIG. 8 is an exemplary H.225 packet;

FIG. 9 and FIG. 10 are exemplary graphical user interfaces according to an implementation of the invention;

FIG. 11 is a flowchart illustrating operation of an implementation of the invention;

FIG. 12A and FIG. 12B are exemplary graphical user interfaces according to an implementation of the invention;

FIG. 13 is an exemplary graphical user interface according to an implementation of the invention; and

FIG. 14 is a diagram of a graphical user interface according to an implementation of the invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGS. 1- 14 illustrate an improved system and method according to the present invention. A first implementation is a system and method for configuring a communication system with system updates without having to recompile associated application programming interfaces. In a second implementation, an H.450 test client is provided. The H.450 test client can send and receive H.450 application protocol data units in any H.225 message. The test client allows for the setting of desired H.450 parameters and allows for the display of the H.450 message itself.

ASN.1 Text Strings

An exemplary computer system implementing a communication system according to an implementation of the invention is shown in FIG. 1. As will be described in greater detail below, the system 102 may be embodied as an H.323 terminal. In particular, the system may be embodied as a personal computer, such as an x86 compatible personal computer or an Apple Macintosh G4. The computer 102 includes a processor 11 adapted to implement computer code according to the present invention. Further,

according to an implementation of the present invention, the processor 11 implements an operating system (not shown) for generating a graphical user interface (GUI) 520 to control system operation.

As shown in FIG. 1, an application program 502, such as a telephony application, one or more APIs 506, and one or more signaling entities 510a-510n are resident. As will be explained in greater detail below, the APIs 506 implement recursion functions 508 that allow ASN.1 text strings to be passed to the application 502 and the signaling entities 510. Thus, as will be explained in greater detail below, when upgrading the system, i.e., when new signaling entities are provided, the APIs need not be altered.

More particularly, FIG. 2 is a functional diagram of the system according to the present invention. The system includes an application program 502, such as a telephony program. The application program 502 communicates via text strings 503a, b with one or more APIs 506, as will be explained in greater detail below. The text strings 503a,b contain the ASN.1 tree structure of the message and the corresponding values.

The ASN.1 tree structure is defined in terms of "valuetrees" and "syntaxtrees." The syntaxtree is static and contains the syntax of all possible information elements of a message structure. The valuetree is dynamic and contains all information elements that exist for a specific message. The value tree also contains the values of the various fields.

In one implementation, the structure of the message, i.e., the syntaxtree is provided to the application program 502 ahead of time. In that case, although the application 502 and the corresponding signaling entity 510 must be updated in order to add new features, the API 506 remains the same.

In an alternate implementation, the structure and syntax of the message are provided to the application program 502 from the stack (not shown). In that case, the application program 502 will first call the corresponding API to retrieve the syntaxtree structure for the message from the stack. In this case, to add new features, both the application program 502 and the API 506 could be left unchanged. The only change then required to

the system is new signaling entity script.

In operation, a first recursive function 508a receives the syntaxtree text strings 503a, fills in the valuetrees, and passes them to one or more signaling entities 510. The signaling entities may be any entities adapted to generate
 5 encodeable data according to specified functions. In particular, the signaling entities 510 may be embodied as H.450 signaling entities. Such signaling entities include the Message Waiting Indication (MWI), Call Transfer, Call Diversion, Call Hold, Call Waiting Call Completion, Call Pickup, and Number Identification signaling entities. The signaling entities 510 may be
 10 implemented in protocol description language (PDL) scripts. Further details on PDL scripts are available from Radvision, Ltd. The PDL scripts contain the state machine code of the signaling entities and also implement machines arranged in a tree structure.

In turn, the signaling entities 510 communicate their respective
 15 APDUs (application protocol data unit) for ASN.1 encoding/decoding 512. The ASN.1 encoded octet strings are then provided onto the network 514.

When messages are received from the network, the ASN.1 octet strings are decoded, and the corresponding APDUs are provided to the appropriate signaling entities 510. The APIs 506 then receive the message
 20 values from the signaling entities 510, interpret them using a second recursion function 508b, and pass them as text strings to the application.

Operation of an embodiment of the invention is illustrated with reference to the flowchart of FIG. 3. In a step 452, an application program 502 builds up a tree-structured text string, i.e., the valuetree, for a message,
 25 such as an H.450 message. In a step 454, the API 506 receives the valuetree string and fills the information elements into the valuetree. In a step 456, the appropriate signaling entity 510 is created according to the operation in the message. In a step 458, the signaling entity sends the message in an H.450 APDU over the network. In a step 460, the H.450 APDU is received
 30 from the network by the receiving party at its corresponding remote signaling entity. In a step 462, the remote signaling entity sends the message's valuetree to its API. In a step 464, the API reads out the valuetree and sends

the information elements to the application program in a tree-structured text string. Finally, in a step 466, the application program receives the tree-structured text string and processes the information elements.

Operation of a particular implementation of the present invention may be illustrated by way of a specific example. According to a specific embodiment, code implementing the example follows as an Appendix. Shown in FIG. 4 is a software model of the example. The model includes an MWI Test Utility 502a, i.e., Application Program. The Message Waiting Test utility 502a tests the H.450 message waiting indication supplementary service. The Message Waiting supplementary service indicates on the served user that messages are waiting on the message center.

The model also includes an H.450 Test API 506a, a Message Center Signaling Entity 510a, a Served User Signaling Entity 510c, and a Message Linker 510b. To signal the relevant information over the network, APDUs with activation, deactivation, and interrogation operations are sent to and received from the network by the Served User Signaling Entity 510c and the Message Center Signaling Entity 510a.

The Message Linker 510b functions to provide a distribution process by determining to which signaling entity a specific text primitive or APDU is to be sent. That is, messages are sent to and received from the signaling entities 510a via the Message Linker 510b.

As seen in the Appendix, the H450 Test API 506a employs the files H450api.c, H450api.h, and H450struct.h. The Supplementary Service employs the files MWI.pdl and MWI-operations.asn. The H450 Test API implements initAPI, setCallback, initH450Root, and endH450Root initialization and termination functions. The initAPI function initializes the API. The setCallback function sets the callback. The initH450Root creates and initializes, and the end450Root function terminates, the H450 root protocol.

As is known, when a PDL state machine is executed, the resulting data structure is called a "process." As the PDL engine executes the state machines, the processes are also arranged in a tree structure and the process tree is built in a hierarchical manner. When a process activates

another process, the new process is located under the existing process in the process tree, the new process being defined as a “child” process of the existing process.

In addition to the process tree, the PDL engine builds another tree called the “protocol” tree. A protocol is a data structure that allows interaction between the application and the PDL machines. The protocol tree has a similar structure to that of the process tree, but not all of the elements of the process tree are recreated in the protocol tree.

The Test H450 API 506a (FIG. 4) implements createH450protocol, closeH450protocol, newH450RootChild, and newH450Child protocol functions. The createH450protocol function creates a new H450 protocol under the protocol root. The closeH450protocol closes the H450 protocol. The newH450RootChild is called when a new protocol is created under the protocol root. The newH450Child function is called when a new child is created under the message linker protocol.

The H450 test API 506a implements the following message functions: sendmessage450, newH450message, fillin, and recursion. The sendmessageH450 function is called when a new message is sent from the application to the H450 protocol. The newH450message is called if the applsend function is executed in the PDL script in order to send a message back to the application.

The fillin() and recursion() function to transform the tree-structured text-strings into valuetree structures and vice versa. The fillin() function receives the string as a parameter and fills in the various fields in the valuetree (XMLstring -> valuetree). The recursion() function reads out the valuetree and parses its nodes and values into a textstring (valuetree -> XMLstring).

FIG. 5 is a flowchart illustrating operation of the embodiment of the invention. In a step 650, the application program builds up a tree-structured text string for an H.450 message. In a step 652, the API receives the text string and fills in the information elements into the value tree using the fillin() function and creates the Message Linker protocol, i.e., the root protocol. In a

step 654, the Message Linker creates the Signaling Entity according to the operation in the message. In a step 656, the Signaling Entity sends the message in an H.450 APDU over the network. In a step 658, the Message Linker receives the H.450 APDU from the network and creates the Signaling Entity according to the operation received. In a step 660, the Signaling Entity receives the APDU from the Message Center and sends its value tree to the API. In a step 662, the API reads the valuetree using the recursion() function and sends the information elements to the application in a tree-structured text string. Finally, in a step 664, the application receives the tree-structured text string and processes the information elements.

As noted above, one aspect of implementing the invention is a graphical user interface. FIG. 6 illustrates an exemplary graphical user interface 520 for the message waiting indication (MWI) test utility. As shown, the graphical user interface 520 includes a message center (MC) test control 524 and a served user test control 526. The GUI 520 further includes a text display region 522, used to display the ASN.1 text strings. The message center test control 524 includes an Activate Request button 528, a Deactivate Request button 530, an Interrogation Response (ack) button 532, and an Interrogation Response (rej) button 534. The served user test control 526 includes an Activate Response (ack) button 536, an Activate Response (rej) button 538, a Deactivate Response (ack) button 540, a Deactivate Response (rej) button 542, and an Interrogation Request button 544. The Message Waiting Indication (MWI) service activates an indication device (e.g. LED) on the served user endpoint if a message (e.g., voicemail) is waiting on the message center. Three operations can be sent over the network: Activation (to activate the MWI (turn on LED)); Deactivation (to deactivate the MWI (turn off LED)); Interrogation (the served user queries its MWI status).

The "XXX Request Buttons" (where XXX stands, e.g., for Activation) request to send the corresponding operation in an H.450 APDU to the peer entity. If the peer entity receives an indication of an operation (e.g. activation), it will respond by either acknowledging the request by pressing

the "XXX Response (ack) Button" or by rejecting the request by pressing the "XXX Response (rej) Button".

In operation, a user can push a button, such as Activate Request 528.

In response, a dialog window 600 (FIG. 7) is displayed. As shown, the dialog window 600 includes a plurality of data fields, and a SAVE button and a SEND button. Shown are servedUserNr, basicService, msgCenterId, nbOfMessages, originatingNr, timestamp, and priority fields. Once filled in, by pressing the SAVE button, the values are stored in a h4507.ini file; by pressing SEND, the message is sent as a text string to the signaling entity.

H.450 Test Utility

As noted above, one aspect of the invention is an ability to send and receive any H.450 protocol data unit (PDU) in any H.225 message. More particularly, FIG. 8 illustrates an exemplary H.225 signaling message 800. As shown, the signaling message 800 includes a TCP header 802 and an H.225.0 V2 message 804. The H.225.0 V2 message 804 is representative of, for example, a SETUP, CONNECT, FACILITY, etc., message. The H.225.0 V2 message 804 includes H.225.0V2/Q.931 information elements 806, setup information element 808 and an H.450 PDU 810. The H.450 PDU 810 includes a network facility extension (NFE) 812, an interpreter APDU (IAPDU) 814, an H.450 feature 816, an MSI extension 818, and an MSI specific feature 820. The NFE 812 defines the type of source and destination of the operation. The IAPDU 814 defines what the receiver is to do if it does not understand the command. The H.450 feature 816 is the APDU itself, and contains an MSI extension which identifies the manufacturer, as does the MSI specific feature 820.

As will be explained in greater detail below, the present invention allows the user to send test H.225 messages, with or without H.450 APDUs. If an H.450 APDU is to be sent, the user can select the specific H.450 APDU and any combination of information elements and network facility extension. The messages are then sent and received, as will be discussed in greater detail below.

A graphical user interface is used to build the H.450 APDU. A graphical user interface in accordance with an implementation of the present invention is shown in FIG. 9. The GUI 900 allows the user to select the H.225 message which is to be tested. For example, the user can select SETUP 904a, ALERTING 904b, or FACILITY 904c buttons to select the corresponding H.225 message. One or more informational fields 902, such as called party and calling party phone number and alias may also be provided. Further, a list field 906 is provided to display the received H.450 message in an XML-like syntax, as generally described above. The list field 906 displays the H.225 message in which the H.450 APDU was received, and also displays all the H.450.1 information elements.

If the user decides to send an H.225 message, e.g., by clicking on SETUP, and after inputting the target number or name, the dialog box 950 of FIG. 10 is displayed. The dialog box 950 allows setting the H.450 information elements. Thus, the dialog box 950 includes an interpretation APDU select dialog 952 and a network facility extension select dialog 954. Using the pulldown 953, the user can select any of a plurality of interpretation APDUs. Alternatively, by deactivating the checkbox 951 the user can elect to send an H.225 message without the IAPDU. Similarly, the user can elect to send no network facility extension, by clicking on the checkbox 955.

The user can input a source and destination entity using the source entity dialog box 956 and destination entity dialog box 958, respectively. The source entity dialog allows the user to enter an entity type 960a and a source entity address 962. The source entity address 962 allows a user to input a value and address type 966a, 968a. As shown, a Party Number type has been selected. The party number dialog 970a allows the user to select a type of party number 972a and then, more specifically, the type of private 974a or public 976a party number. Alternatively, a transport address 978a may be supplied.

Similarly, the destination entity dialog can be used to select an entity type 960b, and includes a destination entity address dialog 964. The destination entity address dialog can be used to input a value 966b and select

a destination entity address type 968b. As shown, a transported type has been selected. The transport ID, e.g., IP address and port, may be input using the dialog 978b. If a party number type had been selected, the user could input a party number type 972b and a private party number type 974b or public party number type 976b.

The entries may be saved by clicking on SAVE 980. Alternatively, the H.225 message may be sent without the H.450 APDU by clicking on button 982 or with the APDU by clicking on button 984.

FIG. 12A and FIG. 12B illustrate graphical user interfaces according to another implementation of the invention. FIG. 12A illustrates a Select H450 APDU Dialog 1200. The Dialog 1200 appears, for example, when Setup (FIG. 9) is clicked. The Select H450 APDU Dialog includes OK button 1202, Cancel button 1204, and New button 1206, as well as List dialog 1208. By clicking on the Cancel button 1204, the user can decide not to send an APDU with this H.225 message, in which case, the Dialog 1200 will close and the message will be sent immediately. Clicking the New button 1206 will start an H450 Builder Application, as will be explained in greater detail below, which allows the building of a new H450 APDU and adding it to the List dialog 1208. After the APDU is created and saved and the user closes the APDU Builder application, the List dialog 1208 is refreshed to show the new APDU.

If the user highlights an APDU in the List dialog 1208 and clicks the OK button 1202, the Select Operation Argument Dialog 1210 (FIG. 12B) appears.

The Select Operation Argument Dialog 1210 includes an OK button 1212, a Cancel button 1214, a New button 1216, and a List Dialog 1218.

By clicking the Cancel button 1214, the user can decide not to send an Argument with this APDU, which will close the dialog and send the message immediately. Clicking on New 1216 will start the H450 Builder Application which allows the user to build a new argument and add it to the List dialog 1218. Once the new Argument is created and the Builder is exited, the List 1218 is refreshed.

If the user highlights an Argument in the List 1218 and clicks the OK button 1212, the H.450 Client processes all necessary RAS procedures and

send the setup message together with the H.450 APDU to the specified destination.

Operation of the H450 Builder is shown in greater detail with reference to FIG. 13 and FIG. 14. More particularly, FIG. 13 illustrates a Create APDU dialog 1300. The Create APDU dialog 1300 includes an OK button 1302, a Cancel button 1304 and an entry dialog 1306. If the user presses Cancel 1304, the application terminates.

Otherwise, at startup, the Builder first asks what type of APDU the user wants to create. The user does so by typing in the appropriate APDU in the Entry dialog 1306. The user can type in "h4501" to build an H.450 APDU or "h450x" to build an argument, where x refers to the number of the H.450 recommendation. When the user clicks "OK" 1302, the Builder GUI (FIG. 14) appears.

The APDU Builder 1400 displays a tree control, which displays the whole syntaxtree of the H450 APDU. By clicking on the nodes of the tree, the user can select or deselect nodes for building. If a leaf node (i.e., an end node) is selected, a dialog box (not shown) appears for the user to fill in data.

For example, FIG. 14 illustrates a syntaxtree for an H4501 supplementary service. The user can select network facility extension, InterpretationAPDU, and serviceAPDU nodes. If the user selects networkfacilityextension, the user can choose a SourceEntity, a SourceEntityAddress, and a DestinationEntity and DestinationEntityAddress. If the user selects destinationEntityAddress, a variety of choices are available, such as H323-ID, url-ID, and the like. The user can select one and fill in the appropriate value.

Turning now to FIG. 11, a flowchart illustrating operation of an implementation of the invention is shown. In step 1000, the user uses GUI 900 (FIG. 9) to select a particular H.225 message for sending. In a step 1002, a selection of the IAPDU may be made, using the checkbox 951 and the dialog 953 (FIG. 10) or the Select APDU, Select Operation Argument dialogs (FIG. 12A-12B). In a step 1004, a selection of the Network Facility Extension may be made using the checkbox 935 and the dialog 954 or the

Builder 1400. As discussed above, this can include selection of a variety of source and destination entities, and addresses, including party numbers or transport IDs. In a step 1006, the user can elect to save the filled in information or send the H.225 message with or without the APDU, using
5 buttons 980, 984, or 984 (or the dialogs 1200, 1210, 1300, 1400). In a step 1005, the H.450 IAPDU is displayed.

The invention described in the above detailed description is not intended to be limited to the specific form set forth herein, but is intended to cover such alternatives, modifications and equivalents as can reasonably be
10 included within the spirit and scope of the appended claims.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202